Intelligent Bacterial Foraging Optimization Technique to Economic Load Dispatch Problem

R. Vijay

Abstract*— *Bacterial Foraging optimization (BFO) is a swarm intelligence technique used to solve problem in power systems. The algorithm is based on the group foraging behaviour of Escherichia coli (E-Coli) bacteria present in human intestine. This social foraging behaviour of E.coli bacteria has been used to solve optimization problems. In this paper, an overview of the biology of bacterial foraging and the pseudo-code that models this process also explained. This paper presents a novel BFO to solve Economic Load Dispatch (ELD) problems. The results are obtained for a test system with three and thirteen generating units. In this paper the performance of the BFO is compared with Genetic Algorithm (GA) and Particle Swarm Optimization (PSO). The results clearly show that the proposed method gives better optimal solution as compared to the other methods.

Index Terms*— *Bacteria Foraging Optimization, Escherichia coli Economic load Dispatch, Genetic Algorithm, Particle Swarm Optimization.

I. INTRODUCTION

Economic load dispatch (ELD) is an optimization problem to find the most economical schedule of the generating units while satisfying load demands and load constraints. The operating cost is reduced by proper allocation of the amount of power [4] to be generated by different generating units.

Nature has been a source of inspiration for the design of several algorithms. One main principle behind nature inspired algorithms is the concept of efficiency, interpreted as the capability of an individual to obtain a sufficient energy source [1-3] in the least amount of time. This procedure called foraging is crucial in natural selection, since the animals with poor foraging strategies are eliminated, and successful ones tend to propagate. Hence, to survive, an animal or a group of animals must develop an optimal foraging policy. Some of the most successful foragers are bacteria like the E.Coli, which employs chemical sensing organs to detect the concentration of nutritive or noxious substances in its environment. The bacteria then moves through the environment by a series of tumbles and runs, avoiding the noxious substances and getting closer to food patch areas in a process called chemotaxis. Besides, the bacteria can secrete a chemical agent that attracts its peers, resulting in an indirect form of communication.

Inspired by the E.Coli foraging strategy [5], in 2002, K. M. passino proposed Bacterial Foraging Optimization Algorithm (BFOA) for distributed optimization and control. BFOA is based on the foraging behaviour of Escherichia Coli (*E.Coli*) bacteria present in the human intestine [6] and already been in use to many engineering problems, BFOA is better than PSO and GA in terms of convergence, robustness and precision.

Swarm optimization methods are very popular in recent days because they have information sharing and conveying mechanisms. Among swarm optimization methods, bacterial foraging [7] is very promising. This method has different set of advantages regarding local minima, randomness, direction of movement, attraction/repelling, swarming and so on. This paper focuses on Bacterial Foraging optimization algorithm to solve the problem of economic load dispatch problem including valve point effects.

II. ECONOMIC LOAD DISPATCH PROBLEM

The concept behind economic load dispatch problem is to minimize the total fuel cost at thermal power plants subjected to the operating constraints of a power system [8]. Therefore, it can be formulated mathematically with a goal function and equality and inequality constraints.

Objective function:

$$Min F_T = \sum_{i=1}^{N_G} F_i(P_{Gi}) = \sum_{i=1}^{N_G} (a_i P_{Gi}^2 + b_i P_{Gi} + c_i) \quad (1)$$

Where

F_T is the total generation cost ($/hr)

F_i is the fuel-cost function of generator i ($/hr)

N_G is the number of generators

P_{Gi} is the real power output of generator i (MW) and

a_i , b_i and c_i are the fuel-cost coefficients of generator

The basic constraints are the real power balance and the real power operating limits

$$\sum_{i=1}^{N_G} P_{Gi} = P_{Load} + P_{Loss} \quad (2)$$

$$P_{Gi}^{\min} \le P_{Gi} \le P_{Gi}^{\max} \qquad i = 1,2 \ldots\ldots N_G \quad (3)$$

Where

P_{Load} is the total load in the system (MW)

P_{Loss} is the network loss (MW) and

P_{Gi}^{min} & P_{Gi}^{max} are the minimum and maximum power generation limits of generator

Generally large turbogenerators have a set of valves at the inlet to the steam turbine. With the increase in power demand these valves are opened sequentially. The throttling loss in a valve is large when it is just opened and is small when it is fully opened.

Manuscript received on April 14, 2012.

R.Vijay, Department of Electrical & Electronics Engineering, Anna University of Technology, Coimbatore, India, Mobile No: +91-9952322511, (e-mail: vijai.mtp@gmail.com).

This is known as valve point loading effect which makes the cost curve highly nonlinear and non convex as shown in Fig. 1.
This phenomenon can be simulated using a recurring rectified sinusoidal function superimposed with the conventional quadratic cost function. Total cost of generation can be expressed as:

$$MinF_T = \sum_{i=1}^{N_G} a_i P_{Gi}^2 + b_i P_{Gi} + c_i + \left| e_i \sin f_i \left\{ P_{Gi}^{min} - P_{Gi} \right\} \right| \quad (4)$$

Where e_i , f_i are constants from the valve point effect of the generating unit *i*.

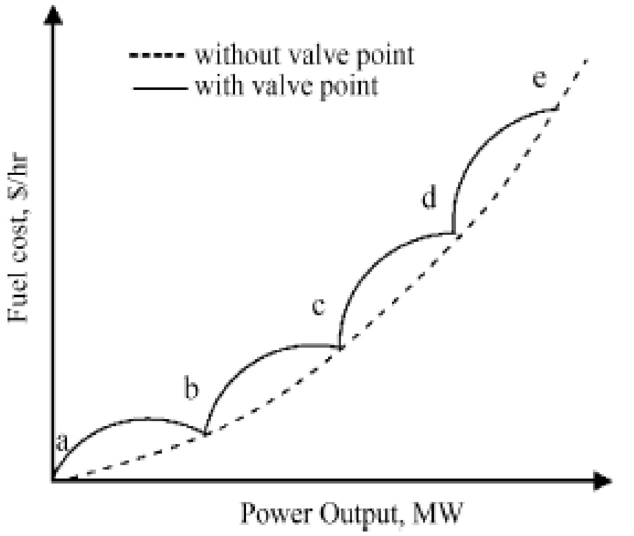

Fig. 1. Input-output curve with valve-point loading. a, b, c, d, e valve points

III. BACTERIAL FORAGING OPTIMIZATION ALGORITHM

BFO method was invented by Kevin M. Passino [12] motivated by the natural selection which tends to eliminates the animals with poor foraging strategies and favor those having successful foraging strategies. The foraging strategy is governed basically by four processes namely Chemotaxis, Swarming, Reproduction, Elimination and Dispersal [14].

A. Chemotaxis

Chemotaxis process is the characteristics of movement of bacteria in search of food and consists of two processes namely swimming and tumbling. A bacterium is said to be 'swimming' if it moves in a predefined direction, and 'tumbling' if moving in an altogether different direction. Let j be the index of chemotactic step, k be the reproduction step and l be the elimination dispersal event. Let $\theta^i(j,k,l)$ is the position of ith bacteria at j^{th} chemotactic step, k^{th} reproduction step and l^{th} elimination dispersal event. The position of the bacteria in the next chemotactic step after a tumble is given by

$$\theta^i(j+1,k,l) = \theta^i(j,k,l) + C(i) \times \frac{\Delta(i)}{\sqrt{\Delta^T(i) \times \Delta(i)}} \quad (6)$$

If the health of the bacteria improves after the tumble, the bacteria will continue to swim to the same direction for the specified steps or until the health degrades.

B. Swarming

Bacteria exhibits swarm behavior i.e. healthy bacteria try to attract other bacteria so that together they reach the desired location (solution point) more rapidly. The effect of Swarming is to make the bacteria congregate into groups and move as concentric patterns with high bacterial density. Mathematically swarming behavior can be modeled as[4]:

$$\begin{aligned} J_{cc}(\theta, p(j,k,l)) &= \sum_{i=1}^{n} J_{cc}(\theta, \theta^i(j,k,l)) \\ &= \sum_{i=1}^{s} \left[-d_{attract} \exp\left(-w_{attract} \sum_{i=1}^{n} (\theta_m - \theta_m^i)^2 \right) \right] + \\ &\quad \sum_{i=1}^{s} \left[h_{repellant} \exp\left(-w_{repellant} \sum_{i=1}^{n} (\theta_m - \theta_m^i)^2 \right) \right] \end{aligned} \quad (5)$$

C. Reproduction

In this step, population members who have had sufficient nutrients will reproduce and the least healthy bacteria will die. The healthier half of the population replaces with the other half of bacteria which gets eliminated, owing to their poorer foraging abilities. This makes the population of bacteria constant in the evolution process.

D. Elimination and Dispersal

In the evolution process a sudden unforeseen event may drastically alter the evolution and may cause the elimination and/or dispersion to a new environment. Elimination and dispersal helps in reducing the behavior of stagnation i.e. being trapped in a premature solution point or local optima.

Pseudo-Code of BFOA
Parameters:
[Step 1] Initialize parameters

$p, S, N_c, N_s, N_{re}, N_{ed}, P_{ed}, C(i)(i = 1, 2, . . . , S), \theta^i$.

Algorithm:
[Step 2] Elimination–dispersal loop: $l = l + 1$

[Step 3] Reproduction loop: $k = k + 1$

[Step 4] Chemotaxis loop: $j = j + 1$

a) For $i = 1, 2, \ldots S$ take a chemotactic step for bacterium i as follows

b) Compute value of cost function $J(i,j,k,l)$
Let
$J(i,j,k,l) = J(i,j,k,l) + Jcc(\theta^i(j,k,l), P(j,k,l))$
(i.e., add on the cell-to-cell attractant effect to the nutrient concentration).

c) Let $J_{last} = J(i,j,k,l)$ to save this value, since we may find a better cost via a run

d) Tumble**:** Generate a random vector $\Delta(i) \in R^p$ with each element $\Delta_m(i), m = 1, 2, \ldots P$ a random number on [-1,1]

e) Move**:** Let

$$\theta^i(j+1,k,l) = \theta^i(j,k,l) + C(i) \times \frac{\Delta(i)}{\sqrt{\Delta^T(i) \times \Delta(i)}}$$

This results in a step of size $C(i)$ in the direction of the tumble for bacterium i.

f) Compute $J(i,j+1,k,l)$ and

let $J(i,j+1,k,l)=J(i,j,k,l)+J_{cc}(\theta^{i}(j+1,k,l),P(j+1,k,l))$

g) Swim

i) Let $m=0$ (counter for swim length)

ii) While $m < Ns$ (if have not climbed down too long)

- Let $m = m+1$
- If $J(i,j+1,k,l) < J_{last}$ (if doing better),

Let $J_{last} = J(i,j,k,l)$

& let

$$\theta^{i}(j+1,k,l) = \theta^{i}(j,k,l) + C(i) \times \frac{\Delta(i)}{\sqrt{\Delta^{T}(i) \times \Delta(i)}}$$

And use this $\theta^{i}(j+1,k,l)$ to compute the new $J(i,j+1,k,l)$ as same in step (f)

- Else, let $m = Ns$. This is the end of the while statement

h) Go to next bacterium $(i+1)$, if $i \neq S$ (i.e., go to [b] to process the next bacterium)

[Step 5] If $j < N_c$, go to step 4. In this case, continue chemotaxis since the life of the bacteria is not over

[Step 6] Reproduction

a) For the given k and l, and for each $i = 1,2,\ldots,S$, let

$$J^{i}_{health} = \sum_{j=1}^{N_c+1} J(i,j,k,l)$$

Sort bacteria and chemotactic parameters $C(i)$ in order of ascending cost J_{health}(higher cost means lower health)

b) The $S_r = S/2$ bacteria with the least healthy bacteria J_{health} values die, and the remaining S_r bacteria with the best values split into two bacteria (this process is performed by the copies that are made are placed at the same location as their parent)

[Step 7] If $k < N_{re}$, go to step 3. If we have not reached the number of specified reproduction steps, so start the next generation of the chemotactic loop

[Step 8] Elimination- Dispersal

For $i = 1,2,\ldots,S$ with probability P_{ed}, eliminates and disperses each bacterium (this keeps the number of bacteria in the population constant)

To do this, if a bacterium is eliminated, simply disperse another one to a random location on the optimization domain or search space. If $l < N_{ed}$, then go to step 1; Otherwise end.

IV. TEST RESULTS AND ANALYSIS

All calculations have been run on Intel(R) Core2Duo(TM) 2.60 GHz CPU, 1 GB RAM, Windows Vista OS and Matlab R2010b compiler. Attributes of the proposed BFOA is done by using trial and error method using following parameters as follows:

No of bacteria S = 10, Number of chemotactic step N_c= 5, No of reproduction step, N_{re}= 10, Elimination dispersal step N_{ed}= 5, probability of elimination dispersal, p_{ed}= 0.25, Ns = 4, depth of the attractant released by the cell $d_{attract}$= 0.1, measure of the width of the attractant signal $w_{attract}$= 0.2, height of the repellent effect $h_{repellent}$= 0.1 & measure of the width of the repellent signal $w_{repellent}$= 1.0 are considered [14] and the test results are compared with the results obtained by other algorithms available in the literature

In order to show the effectiveness of the BFOA, the optimization results for a three unit test system [8] are presented here. Firstly, the algorithm is used to optimize the power dispatch for the problem for two different generating units

A. Test Case I

This test case, adapted from [8,20] comprises three generating units with quadratic cost functions as given in Table I. The dependent-unit active power operating limits were enforced through a quadratic penalty function; after experimentation, a penalty multiplier of 1000 was chosen. The value of population size was taken as 10 for all the test case. The results for best fuel cost for a demand of 850 MW are presented in Table I. The minimum fuel cost of 50 runs for three different algorithms with 100 iterations are carried out to validate the superiority and robustness of the proposed method compared to the other optimization methods[9] are given in Table II. From the results it is clear that BFO gives minimum fuel cost compared to other two.

Table I Units data for test case i (three-unit case) with valvepoint loading

Gen	P_{min} (MW)	P_{max} (MW)	a	b	c	e	f
1	150	600	0.00156	7.92	561	300	0.0315
2	100	400	0.00194	7.85	310	200	0.042
3	50	200	0.00482	7.97	78	150	0.063

Table Iiresults of Test Case I (Three-Unit System) With Load Demand 850 Mw

Parameters	GA	PSO	BFO
P1 (MW)	349.4938	349.4735	349.4662
P2 (MW)	399.258	399.1356	399.198
P 3(MW)	99.924	99.866	99.867
Fuel Cost ($/hr)	8194.98	8194.56	8193.44

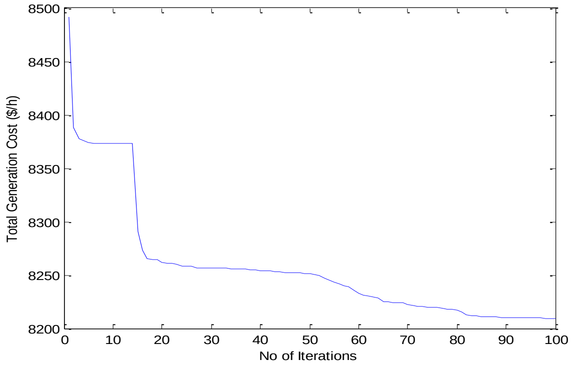

Fig 2. Convergence nature of BFO in test case I (three-unit case).

B. Test Case II

Test case II [3] consisted of 13 generating units as given in Table III. This is a larger system with more nonlinearities. The value of population size was taken as 10 for all the test case.

The value of the penalty factor is chosen as 1000, respectively for better and satisfactory performance in this test case. The results for best fuel cost for a demand of 1800 MW are presented in Table III. The minimum fuel cost of 50 runs for three different algorithms with 100 iterations are carried out. The results obtained from BFO are then compared with GA and from PSO approach from the results shown in Table IV. It is clear that BFO gives minimum fuel cost compared to other two.

Table III Units data for test case ii (13-unit case) with valve-point loading

Gen	P_{min} (MW)	P_{max} (MW)	a	b	c	e	f
1	00	680	0.00028	8.10	550	300	0.035
2	00	360	0.00056	8.10	309	200	0.042
3	00	360	0.00056	8.10	307	200	0.042
4	60	180	0.00324	7.74	240	150	0.063
5	60	180	0.00324	7.74	240	150	0.063
6	60	180	0.00324	7.74	240	150	0.063
7	60	180	0.00324	7.74	240	150	0.063
8	60	180	0.00324	7.74	240	150	0.063
9	60	180	0.00324	7.74	240	150	0.063
10	40	120	0.00284	8.6	126	100	0.084
11	40	120	0.00284	8.6	126	100	0.084
12	55	120	0.00284	8.6	126	100	0.084
13	55	120	0.00284	8.6	126	100	0.084

Table Iv Results Of Test Case Ii (13-Unit System) With Load Demand 1800-Mw

Method	Mean Cost($)	Maximum Cost($)	Minimum Cost($)
GA	18157.67	18214.94	18098.26
PSO	18113.42	18189.83	18054.39
BFO	17997.12	18018.75	17974.48

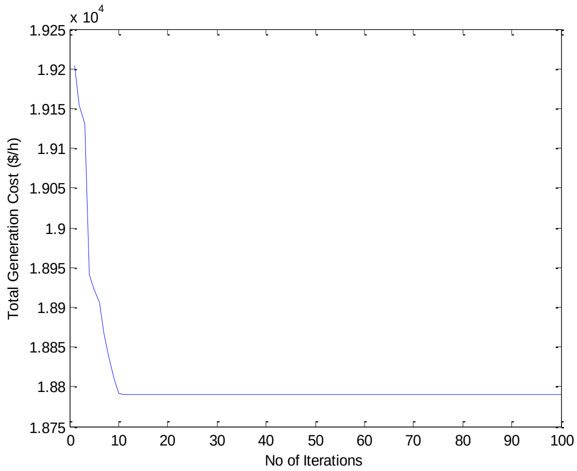

Fig 3. Convergence nature of BFO in test case II (13-unit case).

The Flow Chart for BFO Technique is represented below:

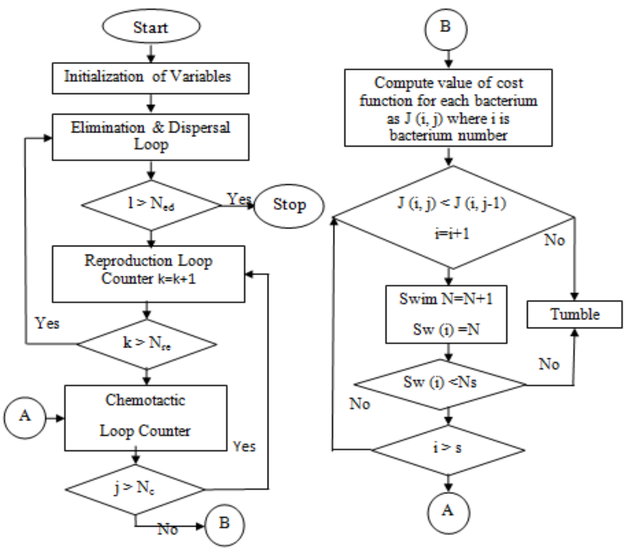

Fig.4. Flowchart of BFO

V. CONCLUSION

In this paper, a novel approach based on BFO has been successfully employed to solve ELD problem including valve point effects.

The proposed algorithm has been tested for a test system with 3 and 13 generating units and the results thus obtained are compared with the results of earlier methods (PSO and GA) available in the literature. As compared to other two, the BFO is easy to implement and there are few parameters to adjust. Therefore, BFO has been successfully applied in many areas of power system. From the outcome of the results, it is shown that the proposed algorithm is very effective in giving quality solutions for ELD problems. Moreover, it also reveals that the fuel costs are reduced.

REFERENCES

1. Kevin M.Passino, “Bacterial Foraging optimization,” International Journal of Swarm Intelligence Research, pp. 1-16, Jan-Mar 2010.
2. Selvakumar, and K. Thanushkodi, "A New Particle Swarm Optimization Solution to Nonconvex Economic Dispatch Problems," IEEE Trans. Power Systems, vol. 22, no. 1, pp. 42-51, Feb. 2007.
3. K.P Wong and J.Yuryevich, "Evolutionary Programming Based Algorithm for Environmentally Constrained Economic Dispatch",IEEE transaction on Power Systems”, Vol. 13,No.2,pp 301,May 1998
4. H. Chowdhury and S. Rahman, "A review of recent advances in economic dispatch," IEEE Trans. Power Syst., vol. 5, no. 4, pp. 1248-1259, Nov. 1990.
5. BijayaKetanPanigrahi,Yuhui Shi, and Meng-Hiot Lim “Handbook of Swarm Intelligence Concepts, Principles and Applications,” Springer pp.487-502, 2011.
6. Jason Brownlee “Clever Algorithms: Nature-Inspired Programming Recipes,” Jason Brownlee pp.257-264, 2011.
7. VeyselGazi and Kevin M. Passino “Swarm Stability and Optimization,” Springer pp.233-249, 2010.
8. J. Wood, and B. F. Wollenberg, “Power generation operation and control,” John Willey and Sons, 2nd Ed, pp. 39-43, 1996.
9. T. Yang, P. C. Yang, and C. L. Huang, “Evolutionary programming based economic dispatch for units with non-smooth fuel cost functions,” IEEE Trans. Power Syst., vol. 11, no. 1, pp. 112–118, Feb. 1996
10. Y. Liu and K. M. Passino, “Biomimicry of social foraging bacteria for distributed optimization: Models, principles, and emergent behaviors,” Journal Optimal. Theory Appl., Vol. 115, no. 3, pp. 603–628, Dec. 2002
11. SibylleD.Muller, JarnoMarchetto, StefanoAiraghi, and PetrosKoumoutsakos,”Optimization Based on Bacterial Chemotaxis,” IEEE Transactions on Evolutionary Computation, vol. 6, no. 1, February 2002.
12. Kevin M.Passino, “Biomimicry of bacterial foraging for distributed optimization and control,” IEEE Control Syst. Mag., Vol. 22, no. 3, pp. 52–67, Jun. 2002.
13. N. Sinha, R. Chakrabarti, and P. K. Chattopadhyay, “Evolutionary programming techniques for economic load dispatch,” IEEE Trans. Evol. Comput., vol. 7, no. 1, pp. 83–94, Feb. 2003.
14. Kevin M. Passino “Biomimicry for Optimization, Control, and Automation,” Springer Verlag London, pp. 768-816, 2005.
15. J.-B. Park, K.-S. Lee, J.-R. Shin, and K. Y. Lee, “A particle swarm optimization for economic dispatch with nonsmooth cost functions,” IEEE Trans. Power Syst., vol. 20, no. 1, pp. 34–42, Feb. 2005.
16. S. Das, A.Biswas, S. Dasgupta, and A. Abraham, “Foundations of Computational Intelligence,”Vol 3: Global Optimization, chapter Bacterial Foraging Optimization Algorithm: Theoretical Foundations, Analysis, and Applications, pages 23–55. Springer, 2009.
17. J. Hazra, A.K.Sinha, “Environmental Constrained Economic Dispatch using Bacteria Foraging Optimization,” IEEE Transactions on Evolutionary Computation,2009.
18. Swagatam Das, S.Dasgupta, A. Biswas, Ajith Abraham, “On Stability of the Chemotactic Dynamics in Bacterial-Foraging Optimization Algorithm,” IEEE Transactions on systems, man, and cybernetics, Vol. 39, No. 3, May 2009.
19. Sambarta Dasgupta,ArijitBiswas,Swagatam Das, BijayaKetanPanigrahi and Ajith Abraham, “A Micro-Bacterial Foraging Algorithm for High-Dimensional Optimization,” IEEE Congress on Evolutionary Computation, pp.785-792, 2009.
20. M. E. El-Hawary and G. S. Christensen, “Optimal Economic *Operation of Electric Power System”,* New York: Academic, 1979.

AUTHORS PROFILE

R.Vijay graduated in Electrical & Electronics Engineering from Bannari Amman Institute of Technology, Sathyamangalam in 2010 and currently he is doing his post graduate in Power systems engineering from Anna University of Technology, Coimbatore. His fields of interest are soft computing techniques in power system and Power System Operation and Control.